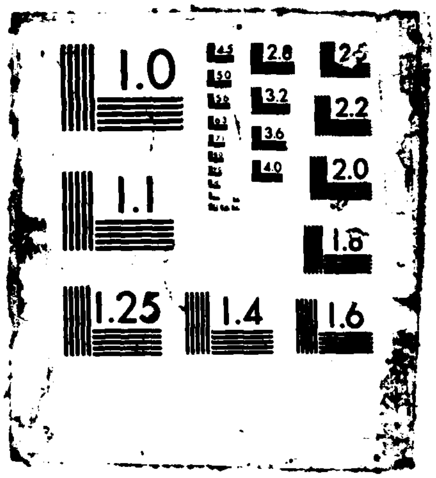
AD-A178 849 A COMPUTER AIDED TOOL FOR SOFTWARE MANAGERS(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF ENGINEERING S L PERALES DEC 86 AFIT/GCS/ENG/86D-4 1/1

UNCLASSIFIED F/G 9/2 NL


1.0
4.5
5.0
5.6
6.3
7.1
8.0
9.0
10
11
12
14
2.8
3.2
3.6
4.0
2.5
2.2
2.0
1.8
1.1
1.25
1.4
1.6

1

A COMPUTER AIDED TOOL FOR SOFTWARE MANAGERS

THESIS

Shirley L. Perales
Captain, USAF

AFIT/GCS/ENG/86D-4

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GCS/ENG/86D-4

A COMPUTER AIDED TOOL
FOR
SOFTWARE MANAGERS

THESIS

Shirley L. Perales
Captain, USAF

AFIT/GCS/ENG/86D-4

AFIT/GCS/ENG/86D-4

A COMPUTER AIDED TOOL
FOR
SOFTWARE MANAGERS

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
In Partial Fulfillment of the
Requirements for the Degree of
Master of Science

Shirley L. Perales, B.S.
Captain, USAF

December 1986

Approved for public release; distribution unlimited

Acknowledgements

Several people deserve credit for the completion of this thesis. First, I would like to thank Captain James Howatt and Captain Wade Shaw for their technical expertise, time, and effort as members of my thesis committee. Second, I want to thank Major Duard Woffinden, my thesis advisor, for his support and encouragement. His willingness to listen and provide sought after advice are much appreciated. Third, I am grateful to my family and friends for their love and moral support. Last, but most important of all, I am indebted to my husband, David. His confidence in me throughout the past 18 months enabled me to "survive AFIT". I couldn't have done it without his love, patience, and support.

TABLE OF CONTENTS

List of Figures

List of Tables

Abstract

The majority of Air Force software development projects are mission critical and usually extremely complex, thus requiring careful management. However, the projects are difficult to efficiently manage due to the abundance of DOD documents governing software development, the DOD effort to standardize regulations for all services, and the lack of adequate tailoring guidelines.

Research was conducted to determine the type of government document and tailoring information useful to a software project manager. Based on the research, a prototype was developed to provide the software manager with information about government documents governing software development. The prototype will provide the information based on the type of project, a given subject or area of interest, and on specific documents. The prototype will also provide tailoring information for the first two steps identified in DOD-HDBK-287.

The prototype demonstrated the feasibility of developing a computer tool to assist the software project manager. This serves as a baseline for future research to determine the feasibility of automating the final two steps in the tailoring process. (Theses).

A Computer Aided Tool for Software Project Managers

I. Introduction

Background

Currently, a multitude of documents describe various aspects of managing a software development contract. The documents governing software development can be categorized as either Department of Defense (DOD) Directives, DOD Instructions, service regulations, or Standards and Specifications. In addition, a variety of other documentation is available to provide guidance to a software manager (14:2-5).

DOD Directives and DOD Instructions govern the military activities for all services, while regulations are service specific implementations of the DOD Directives. Standards and Specifications primarily govern the contractor's effort (14:2-4).

The federal government, DOD in particular, is attempting to standardize software development and acquisition for all government agencies. This results in a need for a standard set of documentation to govern software acquisition and development. The initial document in this set is DOD-STD-2167, "Defense System Software Development" (22:3). This standard is a comprehensive set of requirements for software development, primarily applying to the contractor. However, this standard is not complete since multiple revisions are expected (20:63). The other new and revised documents that form the "2167 package" are (14:2-5):

1. MIL-STD-483A, "Configuration Management Practices"
2. MIL-STD-490A, "Specification Practices"
3. MIL-STD-1521B, "Technical Reviews and Audits"

These documents provide a "maximum" set of requirements that can be levied on a contractor. They should be tailored to fit a project's needs.

Tailoring is the process of determining which portions of a standard are applicable to a project. This can result in a more cost-effective software development effort. DOD-HDBK-287, when written, will provide tailoring guidelines for the "2167 package" (8:1). The tailoring process described in DOD-HDBK-287 is composed of the following four steps:

Step 1 - classify the software by category and use type. Five categories and three use types are described.

Step 2 - select the applicable standards from the "2167 package" and data items.

Step 3 - tailor DOD-STD-2167 requirements, as well as the requirements for the remaining "2167 package" documents.

Step 4 - Tailor the requirements for the selected data items (8:34).

It is difficult to efficiently define the tailoring process since DOD-HDBK-287 is currently in draft form.

The software manager must be aware of and understand all the DOD Directives, DOD Instructions, and service regulations applicable to a specific project. In addition, since a majority of the major software developments are performed by contractors, the software project manager must ensure the contractor abides by the Standards and Specifications governing the contract.

Problem

The majority of Air Force software development projects are mission critical and usually extremely complex, thus requiring careful management. However, the projects are difficult to efficiently manage due to the abundance of DOD documents governing software development, the DOD effort to standardize regulations for all services, and the lack of adequate tailoring guidelines.

Scope

Originally, the thesis was to provide a means of maintaining a project history. However, after interviewing software project managers and software development specialists (15,17,18,24), it was found that commercially available software suitable for this function already exists. Instead, a need for providing information about documents in the "2167 package" and guidance on the tailoring process was identified (15,17,24) and thus, is included within the scope of this thesis. Therefore, the scope of this thesis was modified as follows:

The purpose of this thesis is to design and develop a prototype to provide the software manager with information about government documents governing software development. The prototype will provide the information based on the type of project, a given subject or area of interest, and on specific documents. The prototype will also provide tailoring information for the first two steps identified in DOD-HDBK-287.

Standards

Documentation standards for this research are as specified in AFIT/ENG Development Documentation Guidelines and Standards, Draft #3, dated 22 March 1986.

The human-computer ir.terface is an important aspect of the design. The system will be successful only if the user perceives that useful, accurate information is being provided.

A structured approach to design and coding was used with emphasis on time and space considerations. Structure charts were used in the design phase.

General Approach

This thesis effort required the following actions:

1. Research on the various DOD software development regulations, standards, and directives.
2. Review of the draft document regarding tailoring guidelines for DOD-STD-2167.
3. Design of the databases containing the information from the research activity.
4. Define the operations to be performed on the information contained in the databases.

5. Design an interactive system for the software manager to request information. This includes careful consideration of the human-computer interface and data presentation.

6. Test and debug the code.

7. Evaluate the prototype as specified in the Standards section.

Materials and Equipment

This computer tool was developed on an AT&T 6300 computer system (IBM PC compatible) with a 20 MB hard disk, using the MS-DOS operating system and the TURBO Pascal programming language. No special hardware is required to operate the prototype (e.g., mouse, color monitor). This approach allows the computer tool to be highly transportable.

Sequence of Presentation

Chapter II defines the requirements for the system, describes how the system aids the software project manager, and describes the information to be provided by the system.

Chapter III describes the system design. Design alternatives are presented and discussed.

Chapter IV covers the system implementation, including which design alternatives were chosen and why. Module descriptions are included in this chapter.

Chapter V contains an analysis of the prototype and the results of a preliminary evaluation.

Chapter VI contains the conclusions and recommendations. Results of this research are summarized and suggestions for system enhancements and follow-on research are presented.

II. Requirements Definition

As explained in the previous chapter, the purpose of the system is to provide the software manager with information about government documents governing software development. The purpose of this chapter is to define the requirements for the system. The goal of requirements definition is "to develop a complete, consistent, and unambiguous specification describing what the software product will do, but not how it will do it" (25:150). System requirements can be developed from many different points of view. The requirements definition for this system was developed via a combination of the user's requirements and the system designer's requirements.

This chapter consists of an overview of the type of information the system provides, a database description, the general system specifications, and an introduction to the software engineering tools used.

Information Provided

The system can be functionally divided into two groups based on the type of information needed by the user. The first group consists of all document requests whether for a specific project type, a specific document, or a specific subject. The second group consists of the tailoring guidance for the "2167 package".

Retrieval by Project Type. If the user is not familiar with any of the documents, a full retrieval based on the type of project would be required. The user would select from a list of project types. Enough information to give guidance to the user should be provided. The type of information in this case would be the document number, document name, and a brief description of the document. This information would be repeated for each applicable document.

Subject Retrieval. The user may only be interested in documents pertaining to certain subjects (e.g. quality assurance, test and evaluation). In this situation, a retrieval would be made based on the subject selected. A subject menu would be provided to enable the user to select the subject rather than having to guess at what the system contains. All documents dealing with the subject would be displayed. The same type of document information would be provided as in the previous two retrievals.

Specific Document Retrieval. In the case where the user knows the number of the document, a retrieval is made for the specified document only. A list of the documents currently in the system would be provided to enable the user to select the document rather than entering it. This will help reduce the number of retrieval errors since the input will match exactly. Again, the type of information in this case would be the document number, document name, and a brief description of the document.

Tailoring. An analysis of DOD-HDBK-287, which provides tailoring guidelines for the "2167 package", was conducted to determine which of the four steps in the tailoring process could be incorporated into this project. Steps 1 and 2 are relatively straight forward and are the ones that this prototype will include. Steps 3 and 4 are extensive, requiring the software manager to analyze each paragraph in DOD-STD-2167 in detail.

The user should be allowed to enter tailoring information about a project. The user should be stepped through the decision process for Steps 1 and 2 of the tailoring process. This information would be stored, enabling the user to review the project's information when desired.

An example of the tailoring process is as follows:

Step 1: The user is asked to categorize software by the following category/use types and select those applicable to the particular project:

1. Newly developed software included in a Computer Software Configuration Item (CSCI) - operational.
2. Newly developed software included in a CSCI - support.
3. Newly developed software included in a CSCI - diagnostic.
4. Newly developed software included in a Hardware Configuration Item (HWCI) - operational.
5. Newly developed software included in a HWCI - support.
6. Newly developed software included in a HWCI - diagnostic.
7. Non-deliverable software used in the development environment - operational.

8. Non-deliverable software used in the development environment - support.

9. Non-deliverable software used in the development environment - diagnostic.

10. Unmodified software, either commercially available or reusable, used in a deliverable item - operational.

11. Unmodified software, either commercially available or reusable, used in a deliverable item - support.

12. Unmodified software, either commercially available or reusable, used in a deliverable item - diagnostic.

13. Existing software that will be modified and used in a deliverable item - operational.

14. Existing software that will be modified and used in a deliverable item - support.

15. Existing software that will be modified and used in a deliverable item - diagnostic.

Step 2: The user selects the required standards and data items. DOD-HDBK-287 only addresses the four standards that comprise the "2167 package", therefore, the four selection choices are:

1. DOD-STD-2167, "Defense System Software Development".

2. MIL-STD-483A, "Configuration Management Practices".

3. MIL-STD-490A, "Specification Practices".

4. MIL-STD-1521B, "Technical Reviews and Audits".

The next part of this step is for the user to select the applicable data items. Data item descriptions (DIDs) are the deliverable documentation that a contractor must supply to the government. The need for each data item can be determined based on answers to a series of questions (8:44). The 24 data items to be selected are:

1. System Segment Specification (SSS)
2. Software Development Plan (SDP)
3. Software Configuration Management Plan (SCMP)
4. Software Standards and Procedure Manual (SSPM)
5. Software Quality Evaluation Plan (SQEP)
6. Operational Concept Document (OCD)
7. Computer Resources Integrated Support Document (CRISD)
8. Computer System Operator's Manual (CSOM)
9. Software User's Manual (SUM)
10. Computer System Diagnostic Manual (CSDM)
11. Software Programmer's Manual (SPM)
12. Firmware Support Manual (FSM)
13. Software Requirements Specification (SRS)
14. Interface Requirements Specification (IRS)
15. Software Top Level Design Document (STLDD)
16. Software Detailed Design Document (SDDD)
17. Interface Design Document (IDD)
18. Data Base Design Document (DBDD)
19. Version Description Document (VDD)
20. Software Product Specification (SPS)
21. Software Test Plan (STP)
22. Software Test Description (STD)
23. Software Test Procedures (STPR)
24. Software Test Report (STR)

Data Base

Information on regulations, directives, and standards would be stored in the system's data base as data records. The document records require the document number (unique), name, description, and fields to allow for the retrievals required. As mentioned earlier, the documentation governing software development is currently in a state of flux. Documents are continually being replaced, combined, updated, and added. To accommodate the change in the documentation system, the data base must be easily modified by the user or operator. The following data base functions will allow such modification:

1. INSERT: This function allows the user to insert a new document record into the data base. The system will

prompt the user for the necessary inputs and perform validity checks. Whenever possible, the user will be provided inputs to select from to reduce typing errors.

2. DELETE: This function allows the user to delete an existing record from the data base. The system will prompt the user for the document number, since it is the key attribute. The data record would be displayed to the user to enable verification. The user will be asked to verify that this is the record to be deleted.
3. DISPLAY: This function will display the specified document record to the user. This allows the user to review the document information prior to performing one of the other data base functions.
4. UPDATE: This function allows the user to modify an existing document record in the data base. The system will prompt the user for the document number and display the document information as it currently exists. The user can then update the necessary fields. Again, whenever possible, input selections will be provided to reduce typing errors.

Human-Computer Interface

The human-computer interface is an important aspect of an interactive system such as this prototype. Woffinden (26:41) identifies twelve design principles of which eight were determined to be applicable to this system.

Determine the Purpose of the System. The purpose of this prototype is to provide the software manager with information about government documents governing software development. The prototype will provide the information based on the type of project, a given subject or area of interest, and on specific documents. The prototype will also provide tailoring information for the first two steps identified in DOD-HDBK-287.

Know the User. The target user group for this system is Air Force software project managers. The knowledge and experience level of this group is varied ranging from the novice to the expert. The system designer has three years experience as a software project manager and made several assumptions about the target user group. First, the user can operate the computer on which the prototype will run. Second, the user is not a proficient typist but, is able to use the computer keyboard. Last, the user has access to the government documents referenced by the prototype and is familiar with the acronyms used in software development (e.g., MC for Mission Critical, IS for Information Systems).

Identify Resources Available. The majority of Air Force software project offices have an IBM PC or compatible computer system. Therefore, the primary resource for the implementation of this prototype is the AT&T 6300 computer system, an IBM PC compatible.

Consider Human Factors. The user will normally have few, if any, physical limitations that would interfere with the computer use. Highly developed typing skills should not be required since it is not assumed that the user is a proficient typist. The system is primarily menu-driven, allowing the user to select input options rather than having to memorize the correct commands to operate the system. Data presentation is an important aspect of the system design. The user should be given enough information to produce the correct input responses. This information should be displayed in a clear, concise manner. Headers should be used as a memory aid to let the user know what function is being performed. Information will be represented graphically to provide the user with an easier method of reviewing the data when applicable.

Design for Evolution. The system will be implemented in modules, allowing for easier addition or deletion of sections of code. Graphics routines will be in separate modules since not all IBM PC compatible computers have a graphics capability.

Optimize Training. Since new users will continually be introduced to this system, the amount of training to use the system should be minimized. As previously mentioned, the system is menu-driven allowing a new user to work without the aid of an experienced user.

Use Selection Vs. Entry. Once again, the system is menu-driven allowing the user to select options rather than entering information.

Anticipate Errors. Since the user is assumed to be a non-skilled typist, input errors are to be expected. When inputs are required, validity checks will be performed to reduce the number of invalid system operations.

Timing Considerations

The user should not perceive that he is having to wait an excessive amount of time when using the system. Psychological closure is providing feedback to reassure the user that the system is working which, can be used to help alleviate this problem, but it should only be used if really necessary (25:166). When used excessively, messages will flash across the screen and the user will think he has missed some of the output. Timing should not be a problem since this system is being developed for use on a PC compatible computer system not a larger multiprocessing computer.

Algorithm analysis will be performed using the notation O(n), (big-oh of n), where n represents the number of inputs or outputs, or the number of data items being utilized (21:21). The time order of magnitude of an algorithm refers to the sum of the frequencies of all of its statements (21:22). For example, the frequencies 1, n, and n log n are in increasing order of magnitude. They would be represented as O(1) which is constant, O(n), and O(n log n) respectively. Note that some constant multiplied by n would still be O(n). For example, 2n, 10n, and 100n are all considered to be O(n).

Table I. System Requirements

Requirements	Thesis Prototype	Operational System
Add document record to data base	Interactive	Interactive
Delete document record from data base	Interactive	Interactive
Display document record in data base	Interactive	Interactive
Update document record in data base	Interactive	Interactive
Retrieve document record information by project type	Menu-driven, hard-coded	Menu-driven, use global sets
Retrieve document record information by subject	Menu-driven, hard-coded	Menu-driven, use global sets
Retrieve document record information by specific document	Menu-driven, hard-coded	Menu-driven, use global sets
Step 1 of tailoring process - categorize software by category and use	Allows for 15 combinations, graphics display for output only	15 combinations, graphics for input and output
Step 2 of tailoring process - select required standards	Requires use of DOD-HDBK-287, Interactive	Fully automated, Interactive
Step 2 of tailoring process - select required data items	One CSCI per project, interactive	Multiple CSCIs, interactive
Step 3 of tailoring process - tailor selected standards	not implemented	Required
Step 4 of tailoring process - tailor selected data items	not implemented	Required

Summary

The requirements for the prototype system have been defined and are summarized in Table I. This table compares the requirements for the prototype with those of an operational system.

Having completed the requirements definition for the system, the next chapter discusses the system design.

III. System Design

Based on the requirements definition, this chapter presents the system design. The purpose of the design phase is to take the requirements definition and develop an overall concept of what the system will do. Design alternatives are presented and justification for the chosen design choice is given.

"A structured approach to software design recommends that the overall problem be divided into smaller subproblems which can be solved separately and whose individual solutions, when combined together, give the solution to the original overall problem" (25:173). However, it is difficult to determine the best method of dividing the original problem. This system is divided into modules based on the various functions required. These modules are described later in this chapter.

Structure charts (25:233-235) were chosen to develop the modules for the system design for a number of reasons. First, they are easy to use and understand. Second, structure charts provide a good overall view of the system, showing the modular breakdown of the main problem. Third, structure charts show both control and data flow. Finally, structure charts allow for iterative design, which allows the system to be more flexible to possible design changes. Structure charts have three basic graphic forms: a rectangle represents a module, a vector denotes the control relationships between the modules, and an arrow depicts transfer of data between modules (25:233). The complete set of structure charts for this system is contained in Appendix B, with individual charts being presented throughout this chapter when appropriate.

Main System

The main system should be primarily menu driven allowing the user to select which of the systems' functions is to be executed. The three basic functions are Data Base Manipulations, Document Information Retrieval, and Project Tailoring Information. Figure 1 is the structure chart for this high-level system design.

Data Base Design

Alternatives. Two options were considered for storing the document record information. First, a commercial database could be used. DBase II was evaluated since it was readily available and is applicable for use on an IBM PC or compatible. DBase II is relatively easy to use and could be used to store the required information. However, an interface would be required between it and the implementation language. Such an interface may not exist, requiring the design of one.

The second alternative was to design a database in the implementation language. Use of a commercial data base requires the user to have the data base available, while no additional software is necessary when the data base is

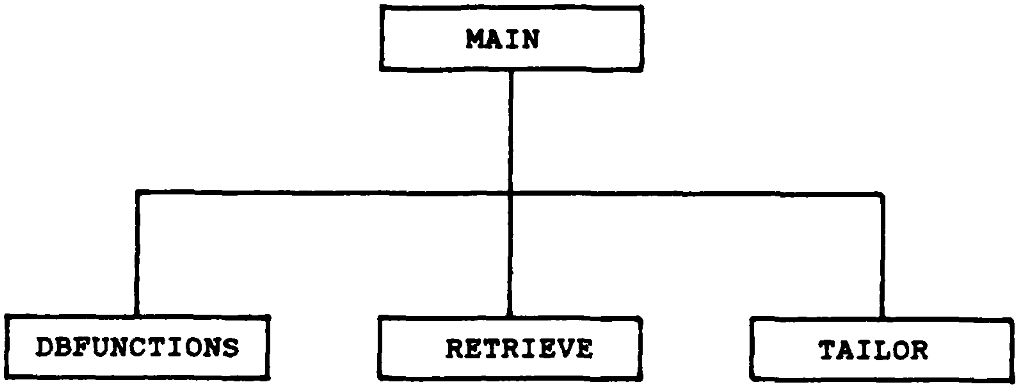


Figure 1. Main Module Structure Chart

designed as part of the tool. This will increase the useability of the tool.

Based on this evaluation, the decision was made to design a data base within the tool.

Data Base Structure. The data base will be comprised of document records. A sample document record is shown in Figure 2.

Number	Name	Description	Subject	Project type

Figure 2. Document Record

The data base functions specified in the Requirements Definition section could either be performed directly on the file or the document records could be read into another data structure for manipulation. Since the file will contain many document records, operating directly on the file will require extensive I/O operations, each composed of the I/O operation itself plus the search time to find the desired record in an unsorted file. However, creating a data structure in main memory requires reading the document records once into the structure and writing the structure back to the file once at the end of the user's operating session.

Use of a data structure was determined to be more efficient for this project. The data structure chosen is the linked list. Linked lists allow for dynamic memory allocation and provide a fast, efficient means of maintaining records. The linked list is created in sorted order by the document number field. The time required to perform a search on the linked list is on average O(n/2) for a list containing n records (18:327). This is approximately the same amount of time required to insert a new document record.

Data Base Functions. The basic data base functions required are INSERT, DELETE, DISPLAY, and UPDATE. The following shows the pseudo code for each function.

```
1.  INSERT (document number)
     Begin
    if HEAD is nil then
        insert as new head record
    else
        move ptr to next record
    while ptr <> nil and input document number >
      ptr document number
        move ptr to next record
    end
    if input document number=ptr document number then
        error - document record exists
    else
        insert new record
     end.
```

```
2.  DELETE (document number)
     Begin
    while input document number <> ptr document number
        move ptr to next record
    end
    if record was found then
        delete record
    else
        error - document record not found
     end.
```

```
3.  DISPLAY (document number)
     Begin
    while input document number <> ptr document number
```

```
            move ptr to next record
        end
        if record was found then
            display record
        else
            error - document record not found
         end.
```

```
    4.  UPDATE (document number)
         Begin
        while input document number <> ptr document number
            move ptr to next record
        end
        if record was found then
            display record
            prompt user for updates
            modify record
        else
            error - document record not found
         end.
```

Since three of the four functions require the location of the desired document record prior to performing its function, it would be useful to have a FIND function. The following is the pseudo code for this function.

```
    FIND (document record)
      Begin
        ptr := head ptr
        while input document number <> ptr document number
            move ptr to next record
        end
        if ptr = nil then
            document record not found
      end.
```

Figure 3 shows the previous system structure chart now modified to depict the data base design.

Document Record Retrieval Modules

The document record retrieval module was subdivided into the three possible document retrieval operations specified in

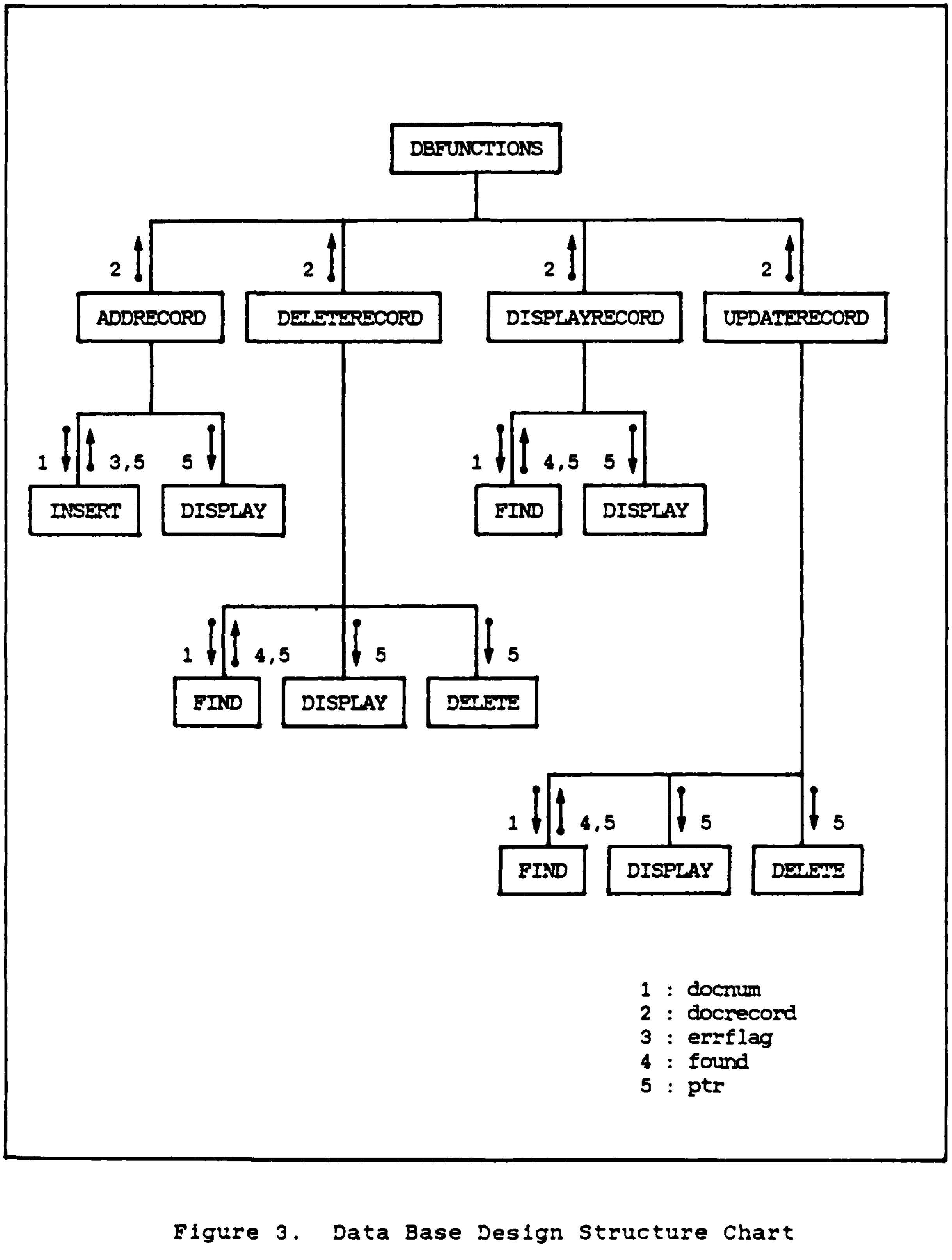


Figure 3. Data Base Design Structure Chart

the requirements definition chapter. Figure 4 shows the structure chart now modified to include the three retrieval module designs.

Project Retrieval Module. The first retrieval module was to obtain the required documents based on the type of project the software is being designed for. This requires being able to designate each document to a particular project type. The three project types identified in most of the documentation used for this research are Mission Critical, Information Systems, and Data Processing. The most efficient means of corresponding a given document with a project type is to include this field as part of the document record in the data base, shown in Figure 2. Notice that each document record can be associated with only one project type. The system would ask the user to select from a menu of project types and a search of the linked list would compare the input project type with the project type field. All documents with the required project type would be displayed for this retrieval. Since the entire list is searched, the time required for this retrieval is O(n). The following shows the pseudo code for this function.

```
PROJECT_RETRIEVAL (project_type)
 Begin
   ptr := head
   while ptr <> nil do
    if ptr project type = input project type then
        display record
        found := true
    move ptr to next record
   end
   if not found then
    message - input project type records not found
 end.
```

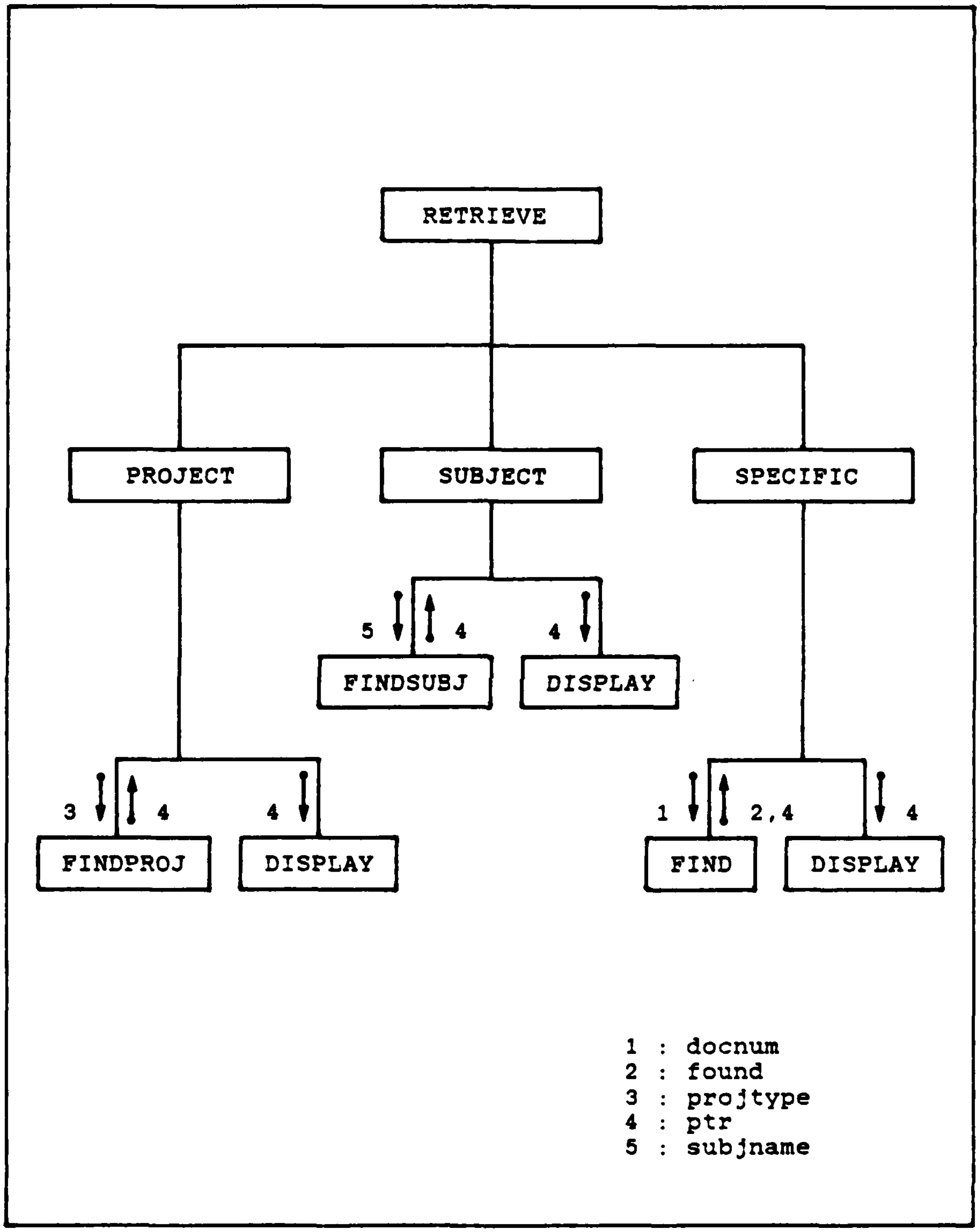


Figure 4. Document Record Retrieval Module Structure Chart

Subject Retrieval Module. The third retrieval module is for the documents pertaining to a given subject area. The user is presented a menu showing which subjects are available for selection. This requires that each document be associated with one of the subjects. A subject field would be included in the document record to allow for this retrieval (see Figure 2). The subjects currently identified are as follows:

1. Configuration Control
2. Design and Development
3. Documentation
4. General Information
5. Quality Assurance
6. Reviews and Audits
7. Software Management
8. Test and Evaluation

The user will select a subject and a search of the linked list is performed matching the input subject with the subject field of the document record. Since all documents corresponding to the subject should be displayed, the entire linked list must be searched. As was the case in the project type retrieval, the time required to perform this retrieval is O(n).

```
SUBJECT_RETRIEVAL (subject)
 Begin
   ptr := head
   while ptr <> nil do
    if ptr subject = input subject then
        display record
        found := true
    move ptr to next record
   end
   if not found then
    message - input subject records not found
 end.
```

Specific Retrieval Module. The second retrieval module required is for a specific document record. The user is presented a menu containing all the document numbers currently in the data base. This allows for selection rather than input on the user's part. The user will select one of the document numbers and the module will retrieve the document record information. In this case, the search is performed matching the input with the document number field. Since this is a one time retrieval with the linked list being searched only until the correct record is found, the time required for this retrieval is on average O(n/2). The following shows the pseudo code for this function.

```
SPECIFIC_RETRIEVAL (document number)
  Begin
     ptr := head
     while ptr document number <> input document number
    move ptr to next record
     end
     display document record
  end.
```

Tailoring Module

The tailoring module should prompt the user for information about the project and then display the results. Steps 1 and 2 from DOD-HDBK-287 (8:34) are included in this computer tool.

The project data must be kept on file so that it is available at a later date. Records used to store each project's information have the format shown in Figure 5.

Project name	Category/use	Standards	DIDs

Figure 5. Project Record

Each project name must be unique since this field will serve as the key field for searches. The "category/use" field has to allow for up to fifteen possible combinations of software categories/use. The "standards" field has to allow for up to four possible standards required for the project. The data item description, "DIDs" field, has to allow for up to 24 possible required DIDs for each project.

The project information will not be accessed to the degree the document information will be. As a result, it will be sufficient to maintain the project records in a file and read in the required project record rather than creating a separate data structure.

Step 1: Classify Required Software. Step 1 asks the user to classify the software by development categories and use type. The system displays the possible category/use combinations and asks the user to make selections. There are fifteen possible combinations the user may select from by combining the five category and three use types (Figure 6).

Since this information can be clearly represented in a matrix format, graphics may be applicable to display the selection information. The system should store this information for the designated project for retrieval at a later time. The system must also be able to display this category/use information when required. Again, graphics should be used to present the information to the user for this step.

	use		
category	operational	support	diagnostic
Newly developed software to be included in a CSCI			
Newly developed software to be included in a HWCI, System or Segment			
Non-deliverable software used in the development environment			
Unmodified software, either commercially available or reusable, used in a deliverable item			
Existing software that will be modified and used in a deliverable item			

Figure 6. Software Category/Use Matrix

Step 2: Select Standards and Data Items. Step 2 asks the user to select the applicable standards from the "2167 package" and the data items required for the software project. DOD-HDBK-287 provides guidance to the user for selection of the standards and data items by asking the user several questions (8:43-48). Three design choices were available for this step. First, the system could prompt the user to answer all of the questions to determine which standards and data items would be required. Second, the system could prompt the user with a subset of these questions and refer the user to DOD-HDBK-287 to provide guidance on how to answer the questions. Third, the system could simply ask whether or not a particular standard or data item were required and rely completely on the use of DOD-HDBK-287 for the user to view the questions.

The second design choice was chosen for a number of reasons. The questions to determine which standards are required are extensive considering only four standards are being decided upon. It would be clearer for the user to read the questions in DOD-HDBK-287 and determine which standards are required for the project. However, the questions to determine which data items are required are straight forward. Twenty-four data items are being decided upon and in most cases, the answer to a single question determines if a particular data item is required. Therefore, selection of the standards is determined by the user using DOD-HDBK-287; the user interactively tells the system which standards are required. The system will determine which data items are required based on an interactive question/answer session with the user. This information must then be stored for each project for retrieval at a later time.

Summary

The system was divided into three functional areas. The data base was designed using linked lists and allows the user to add a document record, delete a document record, display a document record, or update a document record. Each document record must contain fields to allow for information retrievals. A file will be maintained to store the data base information.

Three types of retrievals were designed. The project retrieval module searches the data base and presents the user all of the document records pertaining to the specified project type. The subject module searches the data base and presents the user all of the document records pertaining to the specified subject. The specific document module locates the designated document record in the data base and displays it to the user.

The tailoring module is divided into two submodules. The module to implement Step 1 of the tailoring process presents the user with category/use information and asks the user to select the options applicable to the project. Use of graphics is appropriate for this module. Step 2 of the tailoring process is an interactive session for the user to provide information about the required standards and data items for the project. A separate file was used to maintain the project tailoring information.

Once the system design is complete, the next step is the system implementation which is the discussion of the next chapter.

IV. System Implementation

This chapter describes how the design described in the previous chapter was implemented. Modifications to the original design are highlighted. The implementation hardware, software, and module descriptions will be discussed.

Hardware

The prototype system was developed on an AT&T 6300 Personal Computer (IBM PC compatible) using the MS-Dos operating system. This allows for increased transportability of the prototype since a large number of Air Force offices have IBM PCs or compatible machines. The AT&T 6300 system used has a 20 megabyte hard disk, 640 K RAM, and a graphics capability. The prototype was also tested by running it on the IBM PC AT in the School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, OH, to confirm the system compatibility. The IBM PC AT also has a graphics capability making it compatible with any graphics applications implemented in this prototype.

Software

Having decided on the hardware to develop the prototype, the next step was to select an implementation language. Several options were available such as C, Ada, BASIC, and TURBO Pascal. C would have required researcher learning time to become proficient in the use of the language. The Ada compiler available was only a subset of the Ada language and did not allow for linked list structures, so it was eliminated as a choice. BASIC also would have required researcher learning time and does not lend itself to structured programming, so it was eliminated as a choice. TURBO Pascal was chosen because only a small amount of researcher "refresher" time was required, it had all of the desired application capabilities, and had a graphics package. TURBO Pascal is also similar to Ada, the DOD standard programming language. Therefore, when an Ada compiler with all the desired features becomes available, it will not be difficult to translate the prototype into Ada.

Main System Implementation

The main system begins by calling the procedure BUILD which reads the DOCUMENT.DTA file into the linked list structure. The file is then erased since TURBO Pascal adds data to the end of the file when a write operation is performed. This would result in the data being duplicated each time a write is performed.

The main system displays the main menu to the user. This enables the user to select which of the system functions are to be executed. The main options are:

1. Retrieve Information by Project Type
2. Retrieve Information by Subject
3. Retrieve Specific Document Information
4. Tailoring Information
5. Data Base Modifications
6. Exit

When the Exit option is selected, the main system writes the linked list back to the DOCUMENT.DTA file. Since the linked list is maintained in a sorted order by the document number, the file is also sorted by the document number.

Data Base Implementation

It was possible to implement the data base procedures independently of the other main modules. It was also important to implement it first since the three retrieval modules relied on the data base structure being correct. The data base was implemented using a singly linked list with records as shown below in Figure 7.

Document number: string[17]
Document name: string[75]
Document description: 4 strings[50]
Subject: string[2]
Project type: string[2]
Next: ptr to next record

Figure 7. Linked List Record

A set of procedures were written to perform all of the required manipulations to the linked list itself (i.e., FIND, INSERT, and DELETE). This enables the data structure to be independent of the calling modules so that it could later be replaced with some other structure if desired, or the interface modules could be modified without impacting the structure itself. Modules to perform the data base functions described in the previous chapter were implemented to provide the I/O with the user and to call on the linked list procedures. The following are those procedures that perform the required data base functions.

1. ADDRECORD (documentnum). This procedures allows the user to add a record to the data base. The linked list procedure INSERT is called to determine where to insert the new record. If a document record with the same number already exists, a message is displayed to the user. If the document can be added, the system prompts the user for the remaining input fields required. The DISPLAY procedure is called to display the newly inserted document record.

2. DELETERECORD (documentnum). This procedure allows the user to delete a record from the data base. The FIND procedure is called to locate the record. If the record is not found, an error message is displayed to the user. If the record is found, the DISPLAY procedure is called to ensure the record is actually the one the user wants to delete. A confirmation is required before the record can be deleted. If the confirmation is to delete the record, the DELETE procedure is called to remove the record from the

linked list and deallocate the memory the record occupied.

3. DISPLAYRECORD (documentnum). This procedure displays the desired document record to the user by first locating the record in the linked list with the FIND procedure. If the record is found, it is displayed by calling the DISPLAY procedure. If the record is not found, an error message is displayed to the user.
4. UPDATERECORD (documentnum). This procedure allows the user to update a document record by first locating the record using the FIND procedure. If the record is found, the record fields are displayed individually. The user is given the option of updating a field or leaving it as is. If the record is not found, an error message is displayed to the user.

Once the data base module was working correctly, the retrieval modüles could be implemented and tested. The tailoring modules do not depend on the data base.

Project Retrieval Implementation

The project retrieval module, PROJECT, begins by displaying the current project types available for selection. The user is asked to select one project type and checks are performed to ensure the input was valid. Now, beginning at the head of the linked list, a search is performed comparing the input project type with the project type field in each record. Each time a correct match is made, the document record currently being pointed to is displayed by calling the DISPLAY procedure. This continues through the end of the linked list since this procedure must display all records with the desired input project type.

Subject Retrieval Implementation

The SUBJECT module is very similar to the PROJECT module. The user is presented a menu of the current subject categories to select from and checks are made to ensure the input is valid. Beginning at the head of the linked list, a search is performed comparing the input subject with the subject field of the document records. Each time a match is made, the current document record is displayed via the DISPLAY procedure. This process continues until the end of the linked list is reached. This retrieval procedure is almost identical to the project retrieval except for the retrieval field. However, it was necessary to write separate procedures for these two functions since TURBO Pascal does not allow for procedure calls to be made passing in different parameter types.

Specific Retrieval Implementation

The specific document module, SPECIFIC, begins by displaying the document numbers for all of the documents currently in the data base. This eliminates the need for the user to guess at what documents are available for inspection.

Once the document has been chosen, a search of the linked list is performed. This is done by calling on the linked list FIND procedure. When the document record is found, the DISPLAY procedure is called to display the record to the user. Unlike the project type and subject retrievals, this search and display routine is only performed once since a specific document only exists once in the data base.

Tailoring Module Implementation

This module was implemented independently of the other two main modules since it does not rely on the data base. The tailor records are maintained in the TAILOR.DTA file and are read in only when needed. The tailor record is as shown in Figure 8.

Project Name (unique) : string[15]
Categoryuse : array[15] of boolean
STDs : array[4] of boolean
DIDs : array[24] of boolean

Figure 8. Tailor Record Format

The user is presented a main menu with choices of:

1. Enter New Project Information
2. Review Existing Project Information
3. Exit to the system's main menu

The first two options will now be discussed in detail.

Entering New Project Information. Step 1 requires the user to classify the required software by category and use type. As mentioned in the previous chapter, fifteen possible combinations can be selected from. The matrix previously shown in Figure 6 would have been ideal to display the input choices to the user. The original concept was to display the matrix and ask the user to input the desired category/use. However, TURBO Pascal does not allow for I/O while in the graphics mode. Therefore, the category/use options are displayed in menu format and the user selects from them. Checks are performed to ensure the input is valid. The "categoryuse" field of the tailor record is used to record which options are selected. The default value for this field is a "false"; a "true" value is given to the corresponding array entry when it is chosen.

Step 2 is actually composed of two phases. The first one is to select the required standards for the project. It was previously decided that the user would reference DOD-HDBK-287 to receive guidance on selecting the standards. Therefore, the system simply asks the user to give a "yes/no" response to whether or not a given standard is required. The default value for the "STDs" field is a "false"; if the response is "yes", a "true" value is given to the array entry corresponding to the standard.

The second phase involves selecting the data items required for the project. The system displays a series of questions pertaining to each data item, asking the user to provide a "yes/no" response. Based on the response to these questions, it is determined whether or not a data item is required. The default value for the "DIDs" field is a "false"; a "true" value is given to the array entry corresponding to the particular data item required.

Review of Project Information. The project information is maintained in the TAILOR.DTA file which contains tailor records previously described. The user is asked to input the desired project name for review. The TAILOR.DTA file is searched until a match is made between the input project name and the project name field in the tailor record. If the search is successful, the record is read and available to the system for review. The project data is displayed in the same order it was input (i.e., Step 1 information, Step 2 information). Once again the category/use information could be best represented using a matrix format. This was accomplished using TURBO Basic Graphics which allows for a matrix to be displayed similar to the matrix in Figure 6. The "categoryuse" array for the project record is looped through and an "X" is placed in the corresponding matrix position whenever a "true" value is encountered.

The project's standard information is displayed in a list format. The "STDs" array for the project record is looped through and if the value is "true", the corresponding standard number is displayed.

The project's data item information was implemented similar to the standard information. It also is in a list format. The "DIDs" array for the project record is looped through and if the value is "true", the corresponding data item name is displayed.

V. Analysis and Evaluation

This chapter presents an analysis of the system and discusses the preliminary evaluation conducted.

Analysis of the Prototype System

The document record information contained in Appendix A served as a basis for the initial data base. The document records were added in a random order to ensure that records could be inserted into the beginning, middle, and end of the linked list.

Retrievals were performed on each project type, each document in the data base, and each subject. All retrievals displayed the correct document records or appropriate messages if no document records were available for that retrieval. The information was displayed in a clear, understandable format.

The tailoring modules developed were sufficient to implement the first two steps in the tailoring process as described by DOD-HDBK-287. The question and answer session for determining the data items required seemed rather long, but it is the same process the user would have to go through on paper if this step were not automated. The use of graphics to display the category/use matrix was effective. The user has to know the project's name to be reviewed in the existing system since no data structures are used for the tailor records. This is sufficient when the number of projects in the file are small.

Preliminary Evaluation

A preliminary evaluation of the prototype was conducted to obtain an initial response of the prototype's quality. Six users evaluated the system based on the following seven qualities by using the system and completing a survey.

1. Helpfulness of prompts: range: 1 (not helpful) to 10 (helpful)
2. Clarity of displays: range: 1 (confusing) to 10 (clear)
3. Consistency of displays: range: 1 (non consistent) to 10 (consistent)
4. Usefulness of menus: range: 1 (useless) to 10 (useful)
5. Adequacy of information: range: 1 (not adequate) to 10 (adequate)
6. Ease of use: range: 1 (difficult) to 10 (easy)
7. Usefulness of the prototype: range: 1 (useless) to 10 (useful)

Each user was asked to rate the prototype's qualities based on a low rating of 1 and a high rating of 10. Results of the survey are shown in Table II. Two of the users had software management experience and requested a copy of the prototype since they felt it would be useful in an operational environment.

Table II. System Evaluation

Quality	Average Rating
1. Helpfulness of prompts	8.67
2. Clarity of displays	9.17
3. Consistency of displays	9.67
4. Usefulness of menus	9.33
5. Adequacy of information	8.83
6. Ease of use	9.67
7. Usefulness of the prototype	9.83

. The researcher considers a minimum rating of 5.0 to be sufficient to satisfy the quality. Therefore, the prototype provides a good basis for an operational system since the average ratings are above 8.5 in all cases. The users also provided additional comments, summarized as follows:

1. Provide capability to exit information retrieval review without having to browse all document records (e.g., project type retrieval and subject retrieval).
2. Add a menu level for the retrievals rather than returning control to the system's main menu.
3. Include an exit capability from the tailoring input module.
4. Include a project deletion capability in the tailoring module.

5. Modify error checking to include type checks (e.g., if the user inputs a character when an integer is expected, the system aborts).
6. Clear the screen prior to each input in the ADDRECORD and UPDATERECORD data base functions to increase clarity.

These additional comments are included in the recommendations for system enhancements in the next chapter.

VI. Conclusions and Recommendations

This chapter will summarize the thesis effort and provide recommendations for future research and system enhancements. Included is a summary of the system development, conclusions, and recommendations for system enhancements.

System Development Summary

The system was developed using a Software Life Cycle approach. First, the requirements definition was performed to determine what the system was supposed to do. Next, based on the requirements definition, the system was designed in a repetitive manner using structure charts. Finally, the design was implemented by coding in TURBO Pascal on the AT&T 6300 personal computer system.

The system had three main functions for development. The first was the data base for the document record information. The data base was implemented as a linked list data structure, allowing for easy manipulation of the data without excessive I/O. The linked list procedures were implemented separately from the interface procedures making it possible to use the interface procedures with some other data structure if desired.

The three retrieval functions were developed after the data base was working correctly since it relied on the integrity of the data base. The project retrieval allowed the user to retrieve all the documents pertaining to a particular type of project. The specific document retrieval allowed the user to specify which document was to be displayed. The subject retrieval allowed the user to retrieve all documents pertaining to a specific subject.

The tailoring function implemented Steps 1 and 2 from DOD-HDBK-287. Step 1 requires the user to classify the software by category and use. The system asks the user to select from a menu of category/use combinations and then displays the information in a matrix format using graphics. This display is also used when the project is being reviewed. Step 2 requires the user to specify which standards from the "2167 package" are required for the project and to specify the data items for the project. The system prompts the user as to which standards are required and maintains this data in a tailor record. The system asks the user a series of questions to determine which data items are required. This enables the system to perform the selection process rather than having the user refer to DOD-HDBK-287. Both the standards and data item information are displayed to the user in list format.

Conclusion

This prototype demonstrated the feasibility of developing a computer tool to assist the software project manager. This prototype provided the user with information regarding various government documentation. It also demonstrated the ability to automate portions of the tailoring process that is required for an software project. The original goals of this thesis effort we accomplished successfully.

Re mmendations

The data base is currently only a subset of the do mentation a software project manager may wish to select fr . It can easily be expanded by adding more document re rds. It is recommended that the data base functions be pe ormed by a designated data base administrator since the in grity of the data base is of utmost importance to the re ieval modules. A "save" capability should also be added to th system to enable the data base administrator to save the da base without having to exit the program. Further research sh ld be accomplished to determine if the document record sh ld be modified to allow more than one project type field an or subject field.

The remaining two steps in the tailoring process as sp ified in DOD-HDBK-287 should be evaluated for possible im ementation. These are currently vague areas, but as more in rmation becomes available, it may be feasible to automate th . Further research should be done on evaluating the fe ibility of a data structure for the tailor records if the nu er of projects contained in the TAILOR.DTA file becomes la e. The project names currently in the file could then be di layed allowing the user to make a selection rather than pr ide an input.

The prototype provides useful information as is but, should include the following modifications prior to full operational use:

1. Provide capability to exit information retrieval review without having to browse all document records.
2. Add a menu level for the retrievals rather than returning control to the system's main menu.
3. Include an exit capability from the tailoring input module.
4. Include a project deletion capability in the tailoring module.
5. Modify error checking to include type checks (e.g., if the user inputs a character when an integer is expected, the system aborts).
6. Clear the screen prior to each input in the ADDRECORD and UPDATERECORD data base functions to increase clarity.
7. Protect access to data base.
8. Add "save" capability to data base functions.

After these modifications have been included, the prototype can be sent to software managers in operational offices for further evaluation. The prototype should also be sent to the Air Force Institute of Technology, School of Logistics, Wright-Patterson AFB, OH, for evaluation. Primary emphasis in these evaluations should be on the adequacy of the information presented.

However, the prototype can be used operationally as long as the user is aware of the system's capabilities and uses the user's manual included in Appendix C. Two of the users who participated in the preliminary evaluation requested a copy of the prototype since it will benefit them in their jobs. Also, the system designer will be using the prototype in an operational capacity and continue to make improvements to it.

Bibliography

1. Atre, S. Data Base: Structured Techniques for Design, Performance, and Management. New York: John Wiley & Sons Incorporated, 1980.

2. Bailey, Robert W. Human Performance Engineering: A Guide for System Designers. Englewood Cliffs NJ: Prentice-Hall Incorporated, 1982.

3. Borland. TURBO Pascal, Version 3.0 Reference Manual. Scotts Valley CA: Borland International Incorporated, 1985.

4. Date, C. J. An Introduction to Database Systems (Fourth Edition). Reading MA: Addision-Wesley Publishing Company, 1986.

5. Davis, Richard M. Thesis Projects in Science and Engineering. New York: St. Martin's Press, 1980.

6. Department of Defense. Automated Data Systems (ADS) Documentation. DOD-STD-7935. Fort Belvoir VA: US Army Computer Systems Command, 15 February 1983.

7. Department of Defense. Defense System Software Development. DOD-STD-2167. Washington: Space and Naval Warfare Systems Command, 4 June 1985.

8. Department of Defense. Defense System Software Development Handbook. DOD-HDBK-287 (Draft). Washington: Government Printing Office, 1986.

9. Department of Defense. Development, Acquisition and Management of Mission-Critical Computer Resources (MCCR). DOD Directive 5000.29 (Draft). Washington: Government Printing Office, 1985.

10. Department of Defense. Life Cycle Management of Computer Resources in Systems. AFR 800-14 (Draft). Washington: Government Printing Office, 1986.

11. Department of Defense. Software Quality Assurance Program Requirements. MIL-S-52779A. Fort Belvoir VA: US Army Computer Systems Command, 5 April 1974.

12. Department of Defense. Software Quality Evaluation. DOD-STD-2168 (Draft). Fort Monmouth NJ: US Army ERADCOM, 26 April 1985.

13. Fernandez, Eduardo B. and others. Database Security and Integrity. Reading MA: Addison-Wesley Publishing Company, 1981.

14. Ferrens, Daniel V. Mission Critical Computer Software Support Management. Draft course text. Air Force Institute of Technology (AU), Wright-Patterson AFB OH, 1986.

15. Ferrens, Daniel V., Assistant Professor of Software Management. Personal interview. Air Force Institute of Technology (AU), Wright-Patterson AFB OH, 9 July 1986.

16. Filley, Richard D. "1986 Project Management Software Buyer's Guide," Industrial Engineering: 51-63 (January 1986).

17. Fisher, Steve, GM-14, Deputy Program Director. Personal interview. Air Force Equipment Management System (AFEMS), Wright-Patterson AFB OH, May 1986.

18. Fitzwater, Linda, GS-13. Personal interview. Software Requirements Management Division, AFLC/SITE, Wright-Patterson AFB OH, May 1986.

19. Harwood, Odus E. Software Acquisition and Design Publications within the US Government and the Department of Defense. Research Report. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, March 1986.

20. Heffernan, Henry. "DOD Software Standard Revision to Be Released," Government Computer News, 5 (15): 63 (1 August 1986).

21. Horowitz, Ellis and Sartaj Sahni. Fundamentals of Data Structures in Pascal. Rockville MD: Computer Science Press Incorporated, 1984.

22. Howatt, James. Lecture materials distributed in EENG 545, Software Systems Acquisition. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB, OH, July 1986.

23. Kroenke, David. Database: A Professional's Primer. Chicago: Science Research Associates Incorporated, 1978.

24. Semler, Ron, GS-12. Personal interview. Air Force Equipment Management System (AFEMS), Wright-Patterson AFB OH, May 1986.

25. Woffinden, Duard S. Lecture materials distributed in EENG 593, Software Engineering. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB, OH, September 1985.

26. Woffinden, Duard S. Interactive Environment for a Computer-Aided Design System. MS thesis. Monterey, California: Naval Post-Graduate School, June 1984.

27. Yourdon, Edward. Techniques of Program Structure and Design. Englewood Cliffs NJ: Prentice-Hall Incorporated, 1975.

Appendix A

Documentation Summary

In this appendix the frequently used directives, regulations, standards, specifications, and other documentation for software development are summarized. Other documentation that may be useful to the software manager are listed but not summarized.

AFR 700-9

AFR 700-9 "Information Systems Standards" provides guidance for the acquisition and development of information systems resources and Automated Data Processing Equipment (ADPE).

AFR 800-14

AFR 800-14, Vol I, "Management of Computer Resources in Systems" specifies the responsibilities of Air Force agencies in Mission Critical Computer Resources (MCCR) management. AFR 800-14, Vol II, "Acquisition and Support Procedures for Computer Resources in Systems" is a consolidation of information on the software life cycle, system engineering, testing, configuration management, documentation, and other software related topics from various Air Force regulations and DOD directives governing MCCR management (14:2-1). A revision of AFR 800-14 which combines Vol I and Vol II into one document titled "Life Cycle Management of Computer Resources in Systems" is currently in draft form (10:1-1).

DOD-STD-2167

DOD-STD-2167 "Defense System Software Development" is a joint-service standard governing various aspects of software acquisition and development for MCCR. It is the first attempt at standardizing software development for the Department of Defense and all other Government agencies. This standard can also be applied to non-MCCR software acquisition and development (7:1).

DOD-STD-2168

DOD-STD-2168 "Software Quality Evaluation", currently in draft form, provides detailed guidance for performing quality evaluation of MCCR software and documentation. It can also be applied to software evaluation for non-MCCR. It is intended to be used in conjunction with DOD-STD-2167 (12:3).

DOD-STD-7935

DOD-STD-7935 "Automated Data Systems (ADS) Documentation" is divided into three parts. Part one provides general information on the development and revision of ADS documentation (6:1-1). Part two provides information on the use of DOD-STD-7935 (6:2-1). Part three provides a detailed description of the technical documents and the standards for development of them (6:3-1). DOD-STD-2167 does not affect this standard since this standard governs ADS not MCCR. However, DOD-STD-2167 could be used in place of this standard since it can also be applied to non-MCCR developments.

DODD 5000.29

DODD 5000.29 "Management of Computer Resources in Major Defense Systems" establishes policies for management and control of MCCR. It also provides guidance for less-than-major systems, excluding ADPE (14:2-1). Since this directive also applies to MCCR, it can be used in conjunction with DOD-STD-2167.

MIL-STD-483

MIL-STD-483 "Configuration Management Practices" provides extensive configuration management requirements for hardware and software. This standard should be used when the size or complexity of the software is great, when the software development cycle duration is long, or when many contractors and Government agencies are involved. It is compatible with DOD-STD-2167 although it is not a joint-service standard (8:44).

MIL-STD-490A

MIL-STD-490A "Specification Practices" specifies uniform practices for specification preparation. This ensures that all essential deliverable documentation is included in a contract (8:44). Although not part of the "2167 package", this standard can be used in conjunction with DOD-STD-2167 for MCCR.

MIL-STD-1521B

MIL-STD-1521B "Technical Reviews and Audits" defines precisely what the contractor must present at each review and audit. It is not a joint-service standard, but it is used frequently by all the services. It also is compatible with DOD-STD-2167 (8:44).

The following are other documentation that may be useful to the software manager (19):

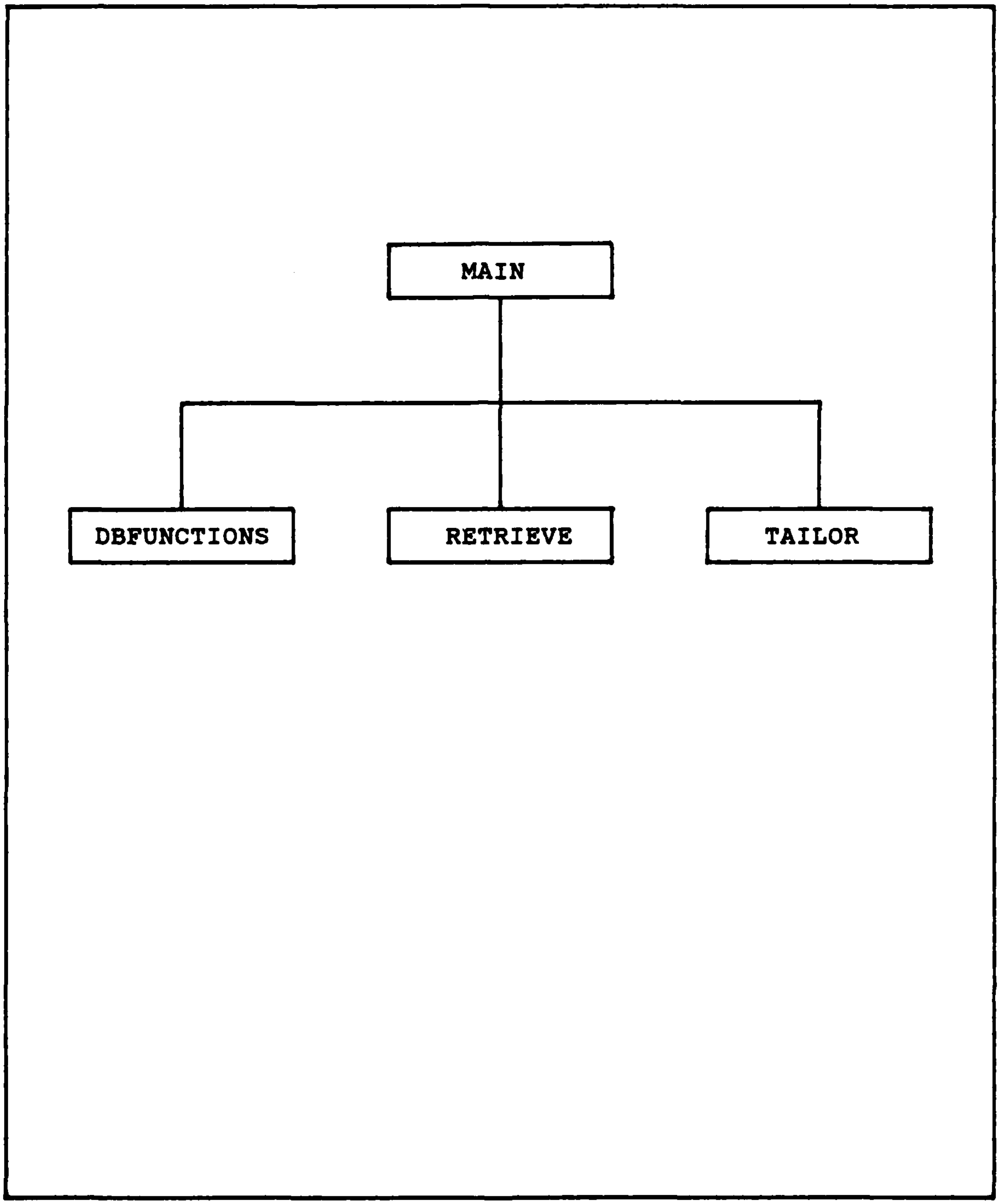
Document Number	Document Title
AFR 65-3	Configuration Management
AFR 74-1	Quality Assurance Program
AFR 74-4	The DOD Quality System Review Program
AFR 74-15	Procurement Quality Assurance
AFR 80-14	Test and Evaluation
AFR 80-24	Reliability and Maintainability
AFR 205-1	Information Security Program
AFR 205-16	ADP Security Policy, Procedures and Responsibilities
AFR 700-1	Managing Air Force Information Systems
AFR 700-2	Information Systems Planning
AFR 700-3	Information Systems Requirements Processing
AFR 700-4, Vol I	Information Systems Program Management
AFR 700-4, Vol II	Information Systems Acquisition and Major Automated Information Systems Review Requirements
AFR 700-10	Information Systems Security
AFR 800-2	Acquisition Program Management
AFR 800-4	Transfer of Program Management Responsibility

Document Number	Document Title
AFR 800-5	Selected Acquisition Reports (SARS)
AFR 800-7	Integrated Logistics Support Implementation Guide for DOD Systems and Equipment
AFR 800-11	Life Cycle Cost Management Program
AFR 800-12	Acquisition of Support Equipment
AFR 800-15	Human Factors Engineering
AFR 800-18	Air Force Reliability and Maintainability Program
AFR 800-19	System or Equipment Turnover
AFR 800-21	Interim Contractor Support for Systems and Equipment
AFR 800-27	Development and Use of Non-Government Specifications and Standards
AFR 800-37	Policies and Procedures for Transition From Development to Production
DOD-STD-480A	Configuration Control-Engineering Changes, Deviations, and Waivers
DOD-STD-1467	Software Support Environment
DOD-STD-1679A	Weapon System Software Development
MIL-HDBK-334	Evaluation of a Contractors Software Quality Assurance Program
MIL-Q-9858A	Quality Program Requirements
MIL-S-52779A	Software Quality Assurance Program Requirements
MIL-STD-109B	Quality Assurance Terms and Definitions
MIL-STD-143B	Standards and Specification, Order of Precedence for the Selection of
MIL-STD-470A	Maintainability Program Requirements (For Systems and Equipment)

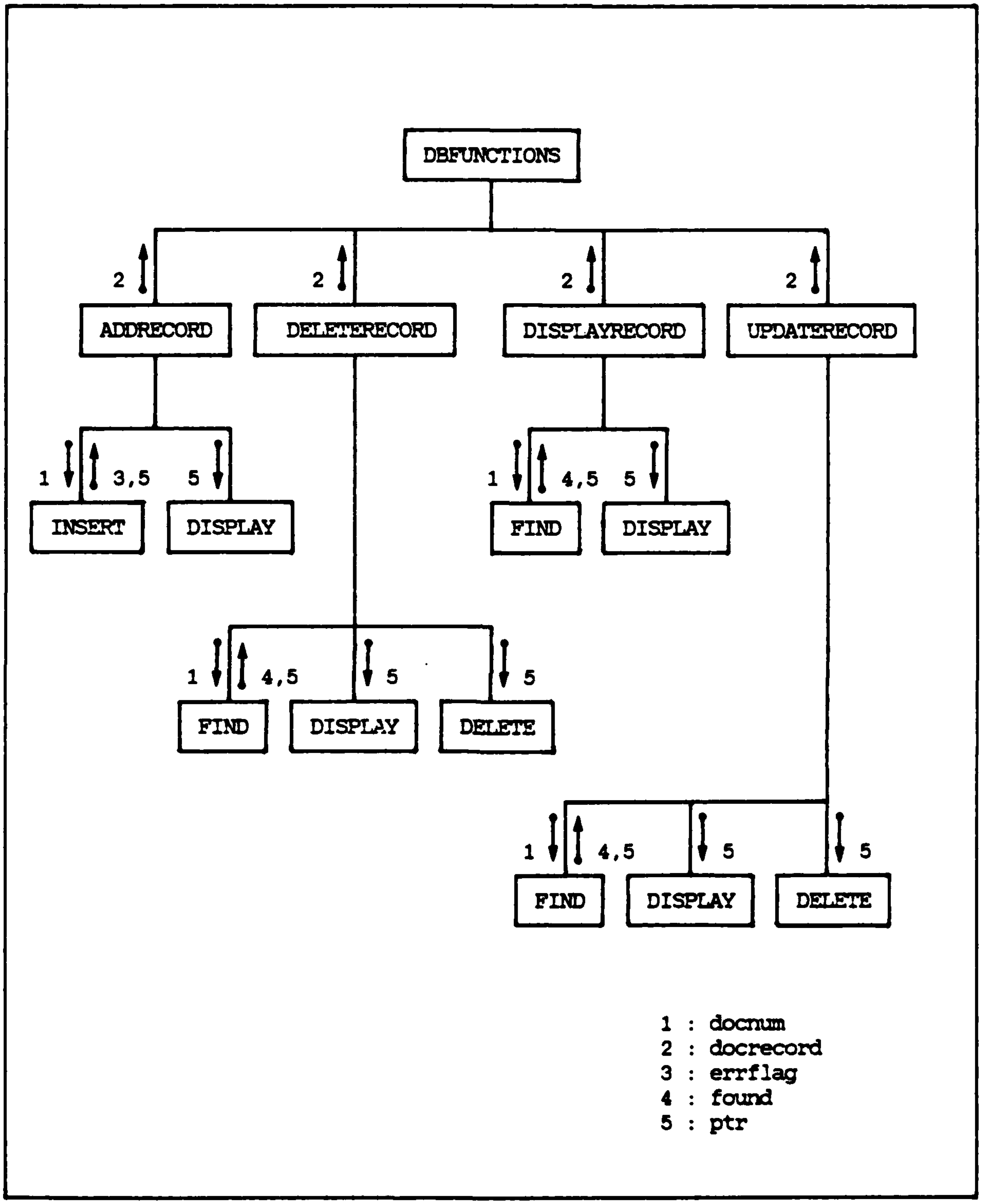
Document Number	Document Title
MIL-STD-481A	Configuration Control-Engineering Changes, Deviations, and Waivers (Short Form)
MIL-STD-482B	Configuration Status Accounting Data Elements & Related Features
MIL-STD-499A	Engineering Management
MIL-STD-721C	Definitions of Effectiveness Terms for Reliability, Maintainability, Human Factors, and Safety
MIL-STD-756B	Reliability Modeling and Prediction
MIL-STD-785B	Reliability Program for Systems and Equipment Development and Production
MIL-STD-1456	Contractors Configuration Management Plans
MIL-STD-1472C	Human Engineering Design Criteria for Military Systems, Equipment, and Facilities
MIL-STD-1815A	Ada Programming Language (ANSI/MIL STD 1815A-1983)
MIL-STD-2068	Reliability Development Tests

Appendix B

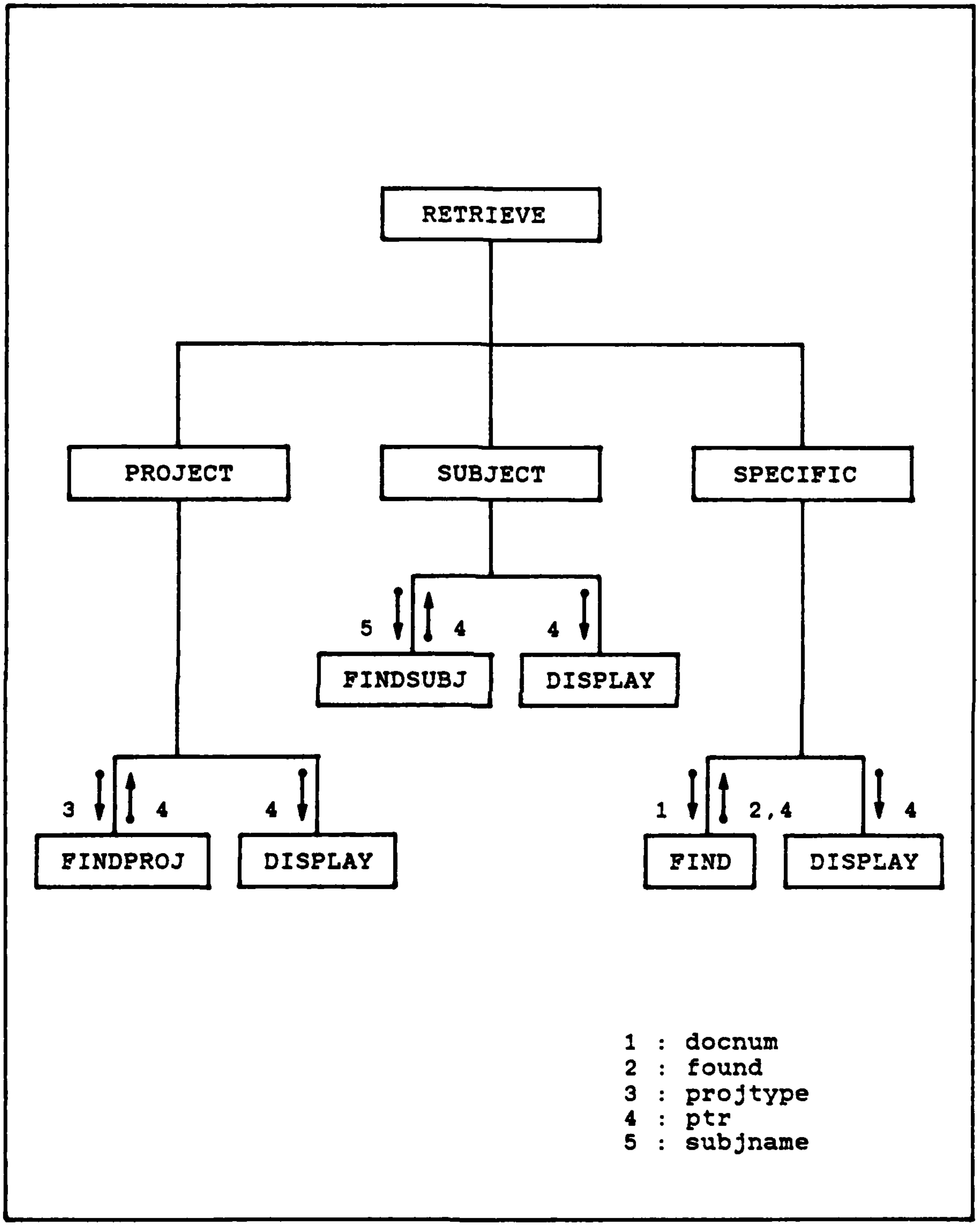
Structure Charts



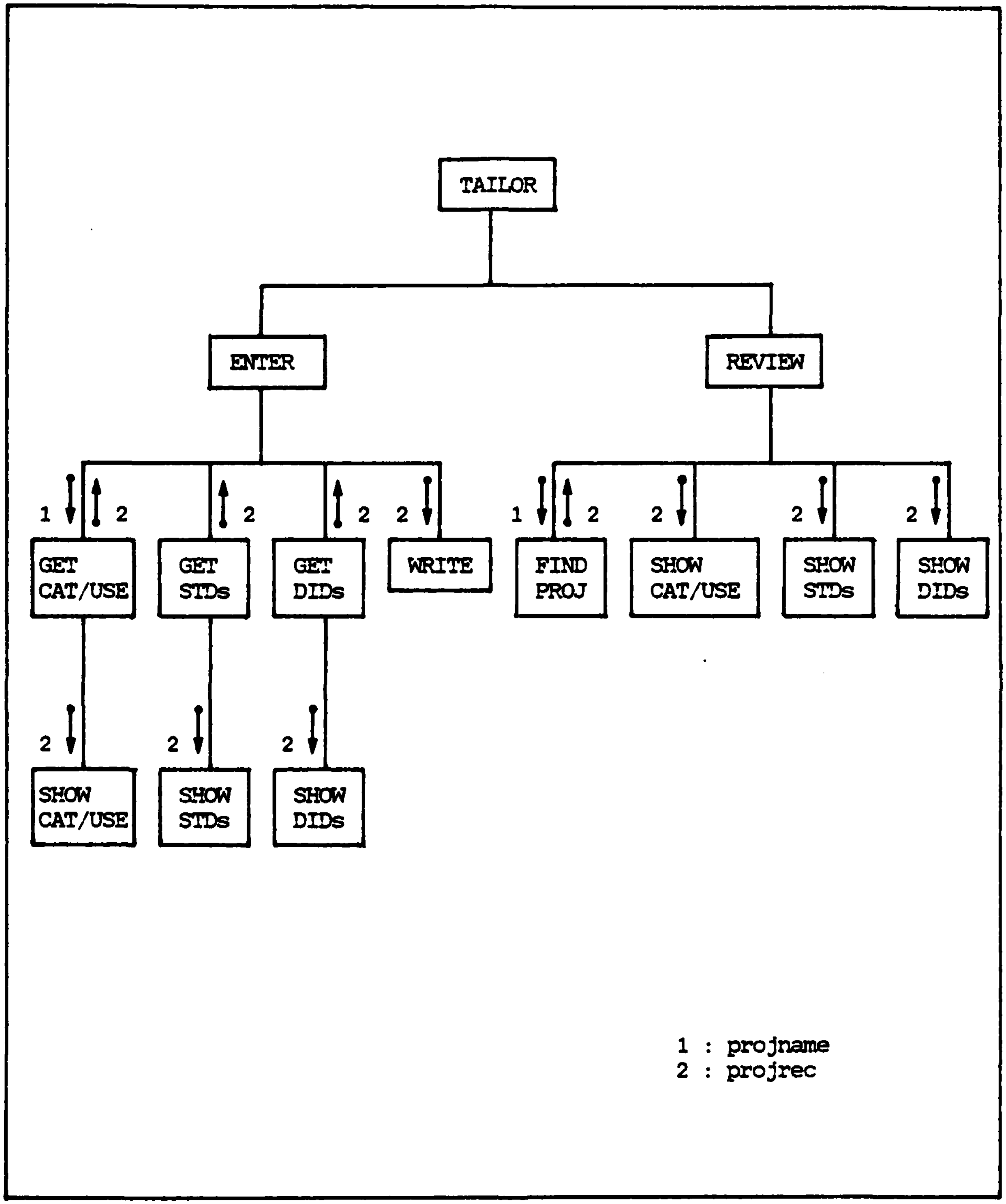
Main Module Structure Chart



Data Base Design Structure Chart



Document Record Retrieval Module Structure Chart



Tailoring Module Structure Chart

Appendix C

User's Manual

This appendix provides a user's manual for the operation of the prototype.

1. The files required for operation are:

   a. THESIS.PAS (source code)
   b. DOCUMENT.DTA (document record data base)
   c. TAILOR.DTA (project tailoring records)

   These files should be loaded on to the system disk if not already present. The TURBO Pascal software must also reside on the system disk.

2. System Start-up:

   a. Type "TURBO" to initialize the TURBO Pascal software.

   b. Type "R" to run the program.

   c. Type "THESIS.PAS <CR>" when asked for the filename to run. This will compile the source code.

   d. The data files are initialized, the data base structure is formed, and the main menu is displayed.

3. System Operations:

   The main menu will appear as follows:

```
                    MAIN MENU

      1:  Retrieve Information by Project Type
      2:  Retrieve Information by Subject
      3:  Retrieve Specific Document Information
      4:  Tailoring Information
      5:  Data Base Modifications
      6:  Exit
```

At the system prompt, enter the option number desired followed by <CR>. After each operation, the main menu reappears until the EXIT option is selected. The following describe the operations by option selected:

a. "Retrieve Information by Project Type" option. The following menu is displayed:

```
                PROJECT TYPES

      MC : Mission Critical
      IS : Information Systems
      DP : Data Processing
```

At the system prompt, enter the option desired (capital letters only) followed by <CR>. The system displays all document records matching the requested project type. To view the next document record, enter <CR> at the system prompt. When the last document record is displayed, the <CR> will return control to the main menu (step 3).

b. "Retrieve Information by Subject" option. The following menu is displayed:

```
                 SUBJECTS

CC : Configuration Control
DD : Design and Development
DO : Documentation
GI : General Information
QA : Quality Assurance
RA : Reviews and Audits
SM : Software Management
TE : Test and Evaluation
```

At the system prompt, enter the option desired (capital letters only) followed by <CR>. The system displays all document records matching the requested subject. To view the next document record, enter <CR> at the system prompt. When the last document record is displayed, the <CR> will return control to the main menu (step 3).

c. "Retrieve Specific Document Information" option.

A menu containing all the document record numbers in the data base is displayed. The document records appear in alphanumeric order. The prototype data base contains nine document records with a menu as follows:

```
        DOCUMENT NUMBERS

    1 : AFR 700 -9
    2 : AFR 800-14
    3 : DOD-STD-2167
    4 : DOD-STD-2168
    5 : DOD-STD-7935
    6 : DODD 5000.29
    7 : MIL-STD-483
    8 : MIL-STD-490A
    9 : MIL-STD-1521B
```

At the system prompt, enter the option number desired followed by <CR>. The system displays the document record matching the input document requested. Enter <CR> at the system prompt to return control to the main menu (step 3).

d. "Tailoring Information" option:

The following menu is displayed:

```
                  TAILORING MENU

      1 : Input Project Information
      2 : Review Project Information
      3 : Exit
```

At the system prompt, enter the option number desired followed by <CR>. After each operation, the tailoring menu reappears until the EXIT option is selected.

1. "Input Project Information" option. Enter the requested information as specified by the system prompts. Control returns to the tailor main menu.
2. "Review Project Information" option. At the system prompt, enter the project name to be reviewed followed by <CR>. Enter <CR> when ready to view the next screen of information. Control returns to the tailor main menu.
3. "Exit" option. Returns control to the system main menu (step 3).

e. "Data Base Modifications" option.

This option should only be selected by the data base administrator. The operational system should provide a means of controlling access to this function (e.g., password protection). The data base main menu is as follows:

```
               DATA BASE FUNCTIONS

      1 : ADD document record
      2 : DELETE document record
      3 : DISPLAY document record
      4 : UPDATE document record
      5 : EXIT
```

Each option is self explanatory by the system prompts provided. Control remains at this main menu until the EXIT option is chosen.

f. "EXIT" option.

The data base is written to the DOCUMENT.DTA file and all files are closed. Enter "Q" to exit TURBO Pascal.

Vita

Capt Shirley L. Perales was born to George and Masako Monhollen on 31 October 1958. She attended Oregon State University, Corvallis, Oregon and received a Bachelor of Science degree in Computer Science in June 1981. Upon graduation, she received a commission into the United States Air Force through the ROTC program. In October 1981, she was assigned to the Air Force Satellite Control Facility, Sunnyvale AFS, California. She performed duty as the Chief of Spacecraft Software Systems in the Vehicle Operations office, VOC, until May 1985. Later that May, she entered the School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, Ohio to obtain a Master's degree in Computer Systems.

Permanent address: 2315 Maple Lane
Steilacoom, WA 98388

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

AD-A178 549

REPORT DOCUMENTATION PAGE

Form Approved
OMB No 0704-0188

1a. REPORT SECURITY CLASSIFICATION
UNCLASSIFIED

1b. RESTRICTIVE MARKINGS

2a. SECURITY CLASSIFICATION AUTHORITY

2b. DECLASSIFICATION / DOWNGRADING SCHEDULE

3. DISTRIBUTION / AVAILABILITY OF REPORT
Approved for public release;
distribution unlimited

4. PERFORMING ORGANIZATION REPORT NUMBER(S)
AFIT/GCS/ENG/86D-4

5. MONITORING ORGANIZATION REPORT NUMBER(S)

6a. NAME OF PERFORMING ORGANIZATION
School of Engineering

6b. OFFICE SYMBOL *(If applicable)*
AFIT/ENG

7a. NAME OF MONITORING ORGANIZATION

6c. ADDRESS *(City, State, and ZIP Code)*
Air Force Institute of Technology
Wright-Patterson AFB, Ohio 45433

7b. ADDRESS *(City, State, and ZIP Code)*

8a. NAME OF FUNDING / SPONSORING ORGANIZATION

8b. OFFICE SYMBOL *(If applicable)*

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER

8c. ADDRESS *(City, State, and ZIP Code)*

10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO	WORK UNIT ACCESSION NO.

11. TITLE *(Include Security Classification)*
A Computer Aided Tool for Software Managers (U)

12. PERSONAL AUTHOR(S)
Perales, Shirley L., Capt, USAF

13a. TYPE OF REPORT
MS Thesis

13b. TIME COVERED
FROM ________ TO ______

14. DATE OF REPORT *(Year, Month, Day)*
1986 December

15. PAGE COUNT
76

16. SUPPLEMENTARY NOTATION

17. COSATI CODES

FIELD	GROUP	SUB-GROUP
05	02	

18. SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*
Data Management
Management Information Systems
Military Publications

19. ABSTRACT *(Continue on reverse if necessary and identify by block number)*

(see reverse)

Approved for public release: IAW AFR 190-1.
Lynn Wolaver 5 Mar 87
LYNN E. WOLAVER
Dean for Research and Professional Development
Air Force Institute of Technology (~~AU~~)
Wright-Patterson AFB OH 45433

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS

21. ABSTRACT SECURITY CLASSIFICATION
UNCLASSIFIED

22a. NAME OF RESPONSIBLE INDIVIDUAL
Woffinden, Duard S., Major, USA

22b. TELEPHONE *(Include Area Code)*
513-255-2057

22c. OFFICE SYMBOL
AFIT/ENG

Previous editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE

The majority of Air Force software development projects are mission critical and usually extremely complex, thus requiring careful management. However, the projects are difficult to efficiently manage due to the abundance of DOD documents governing software development, the DOD effort to standardize regulations for all services, and the lack of adequate tailoring guidelines.

Research was conducted to determine the type of government document and tailoring information useful to a software project manager. Based on the research, a prototype was developed to provide the software manager with information about government documents governing software development. The prototype will provide the information based on the type of project, a given subject or area of interest, and on specific documents. The prototype will also provide tailoring information for the first two steps identified in DOD-HDBK-287.

The prototype demonstrated the feasibility of developing a computer tool to assist the software project manager. This serves as a baseline for future research to determine the feasibility of automating the final two steps in the tailoring process.

END

5-87

DTIC